
Sponge

Version 6.0.0

sept. 07, 2017

Table des matières

1	Introduction	1
2	Sommaire	3
2.1	Les Launchers	3

CHAPITRE 1

Introduction

Bienvenue sur la documentation pour Enoviah. Ici vous pourrez retrouver les éléments qui vous permettrons d'utiliser le plus efficacement nos services. Le monde de la programmation n'étant pas connu de tous, nous mettrons un point d'honneur à rédiger cette docs, afin d'être compréhensible même pour des personnes étrangères au milieu.

Note : Pour l'instant la docs n'est disponible qu'en Français, si vous avez du temps et de solide connaissances en langues n'hésitez pas à nous contacter par mail.

CHAPITRE 2

Sommaire

Les Launchers

Cette section est destinée aux personnes ayant commandés ou souhaitant commander un launcher. Diverses sous-parties vous permettront de comprendre comment utiliser nos launchers, les configurer, etc...

Première Lecture

Enoviah réalise des launchers depuis plus d'un an, fort de son expérience nous vous proposons donc des launchers avec des nombreuses fonctionnalités.

Note : Nous réalisons le code, pas les DESIGNS ! Merci de votre compréhension.

Sommaire

Commander un launcher

Sommaire

Installing Java

Java is needed to run Sponge and Minecraft. You most likely already have Java, but you may need to update it.

Sponge requires Java 8 (specifically 1.8.0_40 or above) at this time. Older Java versions are deprecated and will not work with Sponge. The difference between major versions of Java (6, 7, 8) is significant, and older versions cannot run Sponge properly.

Installing Java

If you have Windows or macOS for your computer, you can [download Java from the official website](#).

Linux users can install OpenJDK via their package manager. OpenJDK is the open source version of the Oracle version of Java, and it should work just as well, if not better. However, it's also possible to [download the Oracle version for Linux](#), but be aware that many Java-dependent Linux packages will still install OpenJDK anyway.

32-bit vs. 64-bit

If your computer supports it, you should use 64-bit versions of Java whenever possible. The Java installers from the linked website should detect whether your computer is ready for 64-bit.

Because the 64-bit version of Java runs considerably better, and also lets Java use more than ~3 GB of your RAM (memory), we always recommend it over 32-bit.

Most modern computers support 64-bit.

JDK vs. JRE

The JRE (Java Runtime Environment) is used to run Java applications. The download page linked above provides the JRE.

The JDK (Java Development Kit) is used to create Java applications, and you do not need it unless you plan to make Sponge plugins or work on Sponge. However, in some cases, you may need the JDK to diagnose a running Java application such as Sponge. You can download the JDK [from a different site](#).

Migrating to Sponge

The purpose of the articles within this section is to help current server owners to migrate from other server platforms to Sponge.

- *Migrating to Sponge*
- *Migrating from CraftBukkit or Spigot*
- *Migrating from Canary*
- *Migrating from Forge*
- *Migrating from Vanilla*
- *Installing Sponge*

Avertissement : Please backup your server files **before** migrating. In case something goes wrong, you still have your backups !

Migrating to Sponge

To migrate to SpongeForge or SpongeVanilla, look at the sections below. Most instructions are identical for both, differences are stated below.

Migrating from CraftBukkit or Spigot

Note : Spigot is a modified fork of CraftBukkit.

Worlds

Forge, and thus SpongeForge (and also SpongeVanilla), use the same world structure as vanilla Minecraft. Vanilla Minecraft places the nether (typically `world_nether`) and the end (typically `world_the_end`) dimensions within the `world` folder. However Bukkit and Spigot *don't* use this system to save the worlds, thus migration is needed.

SpongeForge and SpongeVanilla provide a fully automated conversion script which converts your worlds for you. This is how it works :

1. Shutdown your Bukkit or Spigot server and Backup the entire folder.
2. Install SpongeForge or SpongeVanilla in the folder where you ran your old server, remove the Bukkit or Spigot jars. If you're unsure how you're supposed to install SpongeForge or SpongeVanilla properly, read [this](#).
3. Start the Sponge server, the migrator will be loaded automatically.
4. The Migrator will look into the `bukkit.yml` for a config key called `world-container` and will search that folder for worlds to transfer. If that file isn't around (or something caused it to fail to read), then the migrator will use the `root` folder of the server (which is CraftBukkit standard).
5. Now the migration is performed. At this point, worlds are copied over from what we call the `world container` into the folder defined in the `server.properties` file via the `level-name` key. Note that the original files in the `world container` remain unchanged, a *copy* is made during migration.
6. Bukkit puts things in weird places (with weird names), thus two important fixes have to be applied. Keep in mind that these fixes are based on *assumptions* (due to Bukkit's structure).
 - The first fix is to rename any folder whose name starts with `level-name` property and ends with a Vanilla dimension name (`_nether/_the_end`) to `DIM-1` and `DIM1` respectively.
 - The second fix is to migrate up the `region` data within a Bukkit Vanilla `nether/the_end`. Bukkit puts this data into `DIM-1\region` and `DIM1\region` respectively whereas Vanilla/Forge expect `region` to be in the **root** of the world's folder structure.

The migrator can't provide all needed configuration values. This is the reason why you need to change several parameters by hand to make the world work properly upon loading it on SpongeForge or SpongeVanilla. It is strongly advised to use a *World Management Plugin* to set the right parameters and actually load the world.

Note : As already pointed out, we have to deal with several assumptions while migrating your world. **Thus Sponge does not load the world directly**, you need to install a plugin to handle this.

The output generated by the migration tool should look like this, if everything went fine :

```
[17:32:29] [Server thread/INFO] [Sponge]: Checking for worlds that need to be
↳migrated...
[17:32:29] [Server thread/INFO] [Sponge]: Migrating [world_lol] from [.]
[17:32:29] [Server thread/INFO] [Sponge]: Migrated world [world_lol] from [.] to [
↳\world\world_lol]
[17:32:29] [Server thread/INFO] [Sponge]: Migrating [world_nether] from [.]
[17:32:29] [Server thread/INFO] [Sponge]: Migrated world [world_nether] from [.] to [
↳\world\DIM-1]
[17:32:29] [Server thread/INFO] [Sponge]: Migrating [world_the_end] from [.]
[17:32:29] [Server thread/INFO] [Sponge]: Migrated world [world_the_end] from [.] to
↳[.\world\DIM1]
[17:32:29] [Server thread/INFO] [Sponge]: [3] worlds have been migrated back to
↳Vanilla's format.
```

When this is complete, you should have a copy of the world(s) in a structure that Sponge can load. The original world files remain untouched in their original location, in case something went wrong.

Server and World Configuration Files

CraftBukkit and Sponge both share files that are made available by vanilla Minecraft. These files can thus be reused on Sponge, if they are already present in your CraftBukkit installation :

- `server.properties`
- `banned-ips.json`
- `banned-players.json`
- `ops.json`
- `usercache.json`
- `whitelist.json`

The following files are used by CraftBukkit only, and can be removed because Sponge does not use them :

- `bukkit.yml`
- `commands.yml`
- `help.yml`
- `permissions.yml`

Users who are migrating from Spigot may wish to compare `spigot.yml` to `global.conf` in Sponge. Some keys in `spigot.yml` have counterparts in `global.conf`, and it may be desirable to copy over the values of any keys that are present in both files.

Plugins

Sponge has no native support for Bukkit plugins. However, some members of the community are re-implementing the Bukkit API within a special Sponge plugin, which may allow Bukkit plugins to function on a Sponge server. This plugin has not yet been slated for release.

Ore is Sponge's official repository for finding plugins, and it is recommended to download all Sponge plugins from Ore. When finding replacements for your Bukkit plugins, there are a few points to keep in mind :

- Not all Bukkit developers have chosen to port their plugins to Sponge. Over time, however, someone else may create a suitable replacement.
- Not all Sponge plugins that are ported from Bukkit will automatically convert configuration files. Individual plugin developers make the decision on whether or not to automatically convert configuration files.
- Some Sponge plugins that are ported from Bukkit may change in functionality, or may not even use the same configuration structure.

Migrating from Canary

Worlds

Forge, and thus SpongeForge (and also SpongeVanilla), use the same world structure as vanilla Minecraft. Vanilla Minecraft places the nether (typically `world_nether`) and the end (typically `world_the_end`) dimensions within the `world` folder.

Canary relocates the nether and end dimensions outside of the `world` folder, which must be remedied if it is desired to retain the nether and end dimensions when running Sponge. However, Canary provides an easy method to convert Canary worlds to a structure usable by Sponge with the `/makevanilla` command. If the world conversion is successful, the output will be placed in the `vanilla` folder.

Server and World Configuration Files

Sponge uses many files that are made available by vanilla Minecraft, such as `server.properties`. Canary, however, does not; the only file it has in common with vanilla Minecraft is `usercache.json`. Thus, `usercache.json` is the only file from Canary that can be reused on Sponge.

Nevertheless, it is possible to manually migrate some Canary configuration files to their Sponge counterparts, which have been provided below.

Canary file(s)	Sponge counterpart(s)
<code>server.cfg <world>_<dimension>.cfg</code>	<code>server.properties</code>
<code><world>_<dimension>.cfg</code>	<code>global.conf <dimension>/dimension.conf</code>
<code>ops.cfg</code>	<code>ops.json</code>
<code>db.cfg</code>	No counterpart
<code>motd.txt</code>	No counterpart

Plugins

SpongeVanilla and SpongeForge have no native support for Canary plugins. It may be possible to re-implement the Canary API in a special Sponge plugin.

Ore is Sponge's official repository for finding plugins, and it is recommended to download all Sponge plugins from Ore. When finding replacements for your Canary plugins, there are a few points to keep in mind :

- Not all Canary developers have chosen to port their plugins to Sponge. Over time, however, someone else may create a suitable replacement.
- Not all Sponge plugins that are ported from Canary will automatically convert configuration files. Individual plugin developers make the decision on whether or not to automatically convert configuration files.
- Some Sponge plugins that are ported from Canary may change in functionality, or may not even use the same configuration structure.

Migrating from Forge

Migrating from a plain Forge server to a SpongeForge or SpongeVanilla server is a fairly simple process that needs little-to-no preparatory work.

Migrating to SpongeForge

You must first ensure you are running a version of Forge that is compatible with the version of SpongeForge you plan to use. You may find recommended builds of Forge at [Forge Downloads](#). If you are using any other mods, they must also be updated.

When you are ready to install SpongeForge, you may proceed with the following steps :

1. Stop your Forge server if it is running.
2. Download SpongeForge from the Sponge website and Forge from MinecraftForge.
3. Place `SpongeForge.jar` into your `mods` folder.
4. Start the server and party !

Note : If SpongeForge is the only mod on your server, players will be able to log in with a vanilla client. Other mods may require players to install Forge on their own computers.

Migration to SpongeVanilla

Avertissement : If migrating to **SpongeVanilla** : You will lose all Forge mod data, blocks and entities as SpongeVanilla can't run Forge mods. Keep that in mind when deciding whether you go with SpongeForge or SpongeVanilla.

The process of migration is almost the same as above :

1. Stop your Forge server if it is still running.
2. Download SpongeVanilla and the vanilla server from Mojang.
3. Place your worlds and config files in the server folder.
4. Run the server by launching the `spongevanilla.jar`.

Migrating from Vanilla

Administrators of vanilla Minecraft servers can migrate to Sponge easily because Forge, and thus SpongeForge (and SpongeVanilla), use the same world structure as vanilla Minecraft. Sponge also uses the same files used by vanilla Minecraft, such as `server.properties`.

At first you should decide if you want to run SpongeForge or SpongeVanilla.

Note : Both flavours of Sponge are able to serve vanilla clients. Keep in mind that this only applies to SpongeForge as long as you don't install Forge mods which require client modifications.

1. Stop your Vanilla server if it is still running
2. Download SpongeVanilla or SpongeForge.
3. Place your worlds and config files in the server folder.
4. Run your new server.

Installing Sponge

The guides at [Installing SpongeForge](#) and [Installing SpongeVanilla](#) provide instructions for installing Sponge while you're migrating.

Choosing an Implementation

Something that runs Sponge plugins is called an *implementation*. As long as a plugin is correctly made using the SpongeAPI, it should run correctly on any sufficiently-complete implementation.

Minecraft can't run Sponge plugins out of the box, but you can modify it to do so.

The Sponge API itself is an [open standard](#).

Available Implementations

There are currently two implementations :

Name	Based on
<i>SpongeForge</i>	Mojang's "vanilla" Minecraft and Minecraft Forge
<i>SpongeVanilla</i>	Mojang's "vanilla" Minecraft

Which do I choose ?

If you want to run MinecraftForge mods or you prefer to use Sponge in singleplayer, then choose *SpongeForge*.

If you only want to run a Minecraft server with plugins on it (but no mods), then you can choose *SpongeForge* or *SpongeVanilla*. SpongeForge supports vanilla clients, as long as you don't install Forge mods which require clientside mods. If you prefer to run a server without Forge, then SpongeVanilla is your preferred option.

SpongeVanilla and SpongeForge (without mods) behave the same, so the decision between the two is a matter of preference, not a choice of functionality or features.

Contents

Installing SpongeForge

SpongeForge integrates [Minecraft Forge](#) so you can also run Minecraft Forge mods. In fact, it's more like Sponge itself is a Forge mod that then loads Sponge plugins, but this is a technical detail.

Users who do not want to use Minecraft Forge can consider *SpongeVanilla*.

Download

Grab your copy of [Sponge Forge](#) [here](#).

Reading the Download Filename

When you download SpongeForge, the name of the file will provide some important version information. It includes a Forge build number which this version of SpongeForge is compatible with. Other builds, even ones differing by only a few build numbers are not officially supported.

However, SpongeForge usually updates to a new Forge build fairly soon after it's released, so you needn't worry about always having to run an outdated Forge version in order to use SpongeForge.

The format of the filename is `spongeforge-{MCVersion}-{ForgeVersion}-{SpongeAPIVersion}-{SpongeBuildId}`.

MCVersion	The Minecraft version. Only clients compatible with this version can connect.
ForgeVersion	The version of Forge this file is built for. Preferably your server should run this exact version of Forge.
SpongeAPIVersion	The version of the SpongeAPI implemented by this file. This is what Sponge plugins depend on.
SpongeBuildId	The build number of Sponge. This is what you should supply when reporting bugs or seeking support.

Example

The file name `spongeforge-1.10.2-2254-5.2.0-BETA-2234.jar` is compatible with Minecraft version 1.10.2, requires build 1.10.2-12.18.3.2254 of Forge, provides SpongeAPI 5.2.0 and was build 2234 of SpongeForge.

Note : Normal Forge mods can usually run on any build of Forge for a given Minecraft version (e.g. 1.8.0) without any problems. However, SpongeForge needs to access, among other things, internal parts of Forge, which most mods shouldn't be touching, let alone modifying as Sponge does. Since Forge is free to change internal code whenever they want to, its normal guarantee of backwards-compatibility doesn't apply to SpongeForge.

Installing SpongeForge

Note : If you use (or are planning to use) a game server host, they may have a control panel that can install Sponge for you.

Avertissement : When using the Mojang installer, Mojang makes use of their own Java version and not the one you installed on your system. The installer currently ships with Java 1.8.0_25 for Windows and 1.8.0_60 for macOS. Note that Sponge requires **at least** 1.8.0_40 or above to run properly. You can grab the Launcher without included Java here : [official Minecraft Launcher](#)

Single Player / In-Game LAN Servers

1. Download the Minecraft Forge installer from the [Minecraft Forge website](#). Make sure to use **exactly the same build number** as shown above.
2. Run the provided Forge installer. A new Forge profile will be created in the Minecraft launcher.
3. Open the Minecraft launcher, and select the new Forge profile.
4. Click "Options" and click "Open Game Dir".
5. Download SpongeForge from the Sponge website and put it into the `mods` folder. Create the folder if it does not yet exist.
6. Sponge should work in both in single player and if you open your world to LAN.

Next, learn how you can [configure Sponge](#) and how to manage your instance of Sponge (including installing plugins).

Dedicated Servers

Note : If you already have a Forge server, just put the Sponge mod into your `mods` folder. Remember to update your Forge version to match the one that SpongeForge requires. Have a look at the top of this page if you're unsure which version you need.

Installing Forge via Commandline

1. Visit the [Minecraft Forge website](#) and click “Show all downloads” to view the full set of available options. Identify the version matching the one listed *in the filename of the SpongeForge download*, and hover over the (i) next to “Installer” to get the direct download link.
2. Use your favorite download method to download the jar to its destination. Example : `wget http://url.to/forgedownload.jar`
3. From the folder in which you wish to install Forge, execute the jar with the `--installServer` option. Example : `java -jar forgedownload.jar --installServer`
4. Continue to Adding SpongeForge to Forge below.

Installing Forge via GUI

1. Download the Minecraft Forge installer from the [Minecraft Forge website](#) for the version matching the one listed in the filename of the SpongeForge download. See above for the naming scheme of SpongeForge and Forge.
2. Run the provided Forge installer, select “Install Server”, choose an empty folder to place the server’s files, and then click OK.
3. Continue to Adding SpongeForge to Forge below.

Adding SpongeForge to Forge

1. Download SpongeForge from the [Sponge website](#) and put it into the `mods` folder in your server directory. Create the folder if it does not yet exist.
2. You may now launch the server via command or launch script `java -jar forge-version-XYZ.jar`.
3. If operating from home, set up [Port Forwarding](#) to ensure others can connect.

Next, learn how you can create and use a [launch-script](#), [configure Sponge](#) and manage your server (including installing plugins).

Links

- [Homepage](#)
- [GitHub](#)

Installing SpongeVanilla

SpongeVanilla is a vanilla implementation of the Sponge API as a stand-alone server.

Overview

SpongeVanilla is an implementation of the Sponge API that is created by patching the vanilla Minecraft server. This means it is a stand-alone server, and does not utilise nor require Minecraft Forge or Forge mod loader (FML). SpongeVanilla is being developed in parallel to the Forge version of Sponge, as an alternative platform for users who do not want to run a Forge server. Originally started as an independent project and named Granite, by developers **Azureus-Nation** and **VoltaSalt**, the SpongeVanilla team officially joined the Sponge development team in March 2015.

Download

Grab your copy of [Sponge Vanilla](#) here.

Installing SpongeVanilla

SpongeVanilla only works as a dedicated server.

1. Download the `SpongeVanilla.jar` from the [Sponge website](#).
2. Run it via command line : `java -jar spongevanilla-whatever.jar`
3. Set up [Port Forwarding](#) to ensure others can connect.

Avertissement : Don't double-click the `.jar` file !

Links

— [GitHub](#)

Creating a Launch Script

Note : These instructions apply only if you plan to run your Minecraft server on your own machine. Most shared Minecraft hosts will create a launch script for you.

Writing a Launch Script

First, open a text editor such as Atom, Sublime Text, or Notepad. Write (or paste) a launch script for your server. Examples of simple launch scripts for Windows, macOS, and Linux have been provided below. Keep the RAM limitations of your machine in mind.

Note : The following examples are generic. For a Forge server using Sponge (coremod), change `forge-1.8-XYZ-universal.jar` to whatever your Forge version in the server directory is named. To launch a SpongeVanilla server, change `forge-1.8-XYZ-universal.jar` to the name of the `SpongeVanilla.jar` file.

Windows

```
java -Xms1G -Xmx2G -jar forge-1.8-XYZ-universal.jar
pause
```

Save your Windows launch script as `launch.bat`.

macOS

```
#!/bin/bash
cd "$(dirname "$0")"
java -Xms1G -Xmx2G -jar forge-1.8-XYZ-universal.jar
```

Save your Mac launch script as `launch.command`.

Linux

```
#!/bin/sh
cd "$(dirname "$(readlink -fn "$0")")"
java -Xms1G -Xmx2G -jar forge-1.8-XYZ-universal.jar
```

Save your Linux launch script as `launch.sh`.

Running a Launch Script

Ensure you are running your launch script out of a folder created especially for your server. This is for your own sanity; unfortunately, Sponge is unable to soak up your tears if you do not do this.

You may run your launch script by double-clicking it. If you are using a console or terminal, navigate to the directory of the script and run it. Keep in mind that you must agree to the Mojang EULA in order to run a server.

Note : The default Minecraft server GUI console is disabled by Sponge, because it is very processor-intensive.

Avertissement : If you get a permissions error when attempting to launch your server on a Mac, try this :

- Open the Terminal.
- Type `chmod a+x`, with a space at the end.
- Drag your launch script to the Terminal.
- Press enter.

Port Forwarding

If you are running your Sponge server from your home, it is necessary to set up Port Forwarding on your modem or router in order for other people to connect.

Avertissement : Ensure you take the necessary precautions when port forwarding, because it can be insecure.

Minecraft, and thus Sponge, uses port 25565 by default. Therefore, port 25565 must be port forwarded to the internal IP address of your computer. UDP and TCP are the protocols that must be forwarded.

Astuce : You may change which port is used by editing the appropriate key in your *server.properties*.

Port Forwarding can be performed through your router's administrator panel. If you do not know how to navigate to your router's administration panel, conduct an internet search for specific instructions on port forwarding for your router. The instructions are typically different for every router.

If it is preferable to have your server online upon starting up, you may need to port forward your hardware.

Using Sponge with BungeeCord

BungeeCord is a piece of server proxy software written by md_5 and the SpigotMC team that allows server owners to link Minecraft servers together so that players can jump between servers without having to disconnect and re-connect. BungeeCord is typically used by server networks that offer many game modes.

For more information about BungeeCord, what it is, how to set it up and how it works, have a look at the [BungeeCord website](#). This page will focus on Sponge specific steps.

Avertissement : In order to connect servers to BungeeCord, you must run the servers in offline mode. In offline mode, without the proper precautions, anyone can log into the server using any name they wish, including those who have admin permissions. Make sure you protect your servers using firewalls. If you are using linux, there is an IPTables guide at [SpigotMC Firewall guide](#), alternatively, some distributions come with [UncomplicatedFirewall](#) “ufw”.

If you are not comfortable with tinkering with Linux, or you are unsure as to how to prevent unauthorised access to your servers, consider consulting with someone who has more experience to ensure the security of your server.

Note : Be sure that if you use SSH to make sure port 22 is ALLOWED, otherwise you run a very real risk of locking yourself out of your server !

IP Forwarding

BungeeCord has a mode called IP Forwarding, which allows BungeeCord to pass the player's UUID and IP address to any connected server, even though the servers are being run in offline mode. With current builds of BungeeCord, IP Forwarding works with SpongeVanilla, whilst IP Forwarding only supports SpongeForge when vanilla clients connect - modded servers that require modded clients cannot natively make use of IP Forwarding with the current version of BungeeCord. SpongeForge is only fully supported with the use of a patched version of BungeeCord, or a community supplied BungeeCord plugin.

A pull request has been supplied to BungeeCord to allow BungeeCord to support SpongeForge natively. We are awaiting it to be included in the main product :

- Old PR, has context : [BungeeCord PR 1557](#)
- New PR, uses a different method to avoid breakages : [BungeeCord PR 1678](#)

Using BungeeCord without IP Forwarding

While it is recommended that you use IP Forwarding wherever possible. If you do not wish to do so, simply ensure that `online-mode` is set to `false` in your `server.properties` file and add the server details to Bungee's `config.yml` file. Bungee will then forward any connections to the server when required. It is a good precaution to set the `server-port` to something other than 25565.

This will work with all implementations of Sponge, including with mods.

Using BungeeCord with IP Forwarding

If you wish to use IP Forwarding :

- In the BungeeCord `config.yml`, set `ip_forward` to `true`
- In Sponge's config (`config/sponge/global.conf`), set `modules.bungeecord` to `true` and `bungeecord.ip-forwarding` to `true`
- If you have any other server software, consult the documentation for that server.

This must be done for **all** servers that are connected to the BungeeCord network. Then, just follow the instructions for using BungeeCord without IP Forwarding.

Configuring Sponge

You can find all configuration files inside the “config” folder.

Config Syntax

Most configuration files will make use of the *HOCON* format.

Introduction to HOCON

HOCON (Human-Optimized Config Object Notation) is an easy-to-use configuration format. It is used by Sponge and individual plugins utilizing the Sponge API to store important data, such as configuration or player data. HOCON files typically use the suffix `.conf`.

Components

- a key is a string preceding a value
- a value is a string, number, object, array, or boolean following a key
- a key-value separator separates keys from values, and can be either `:` or `=`
- a comment is prefixed with `#` or `//`, typically serving to provide feedback or instructions

Example :

```
yellow-thing: "Sponge"
```

In this example, the key is `yellow-thing`, the value is `Sponge`, and the key-value separator is `:`.

Working with HOCON

HOCON is more flexible than the JSON (JavaScript Object Notation) format in that there are several ways to write valid HOCON. Below are two examples of valid HOCON.

Example #1 :

```
player: {
  name: "Steve",
  level: 30
}
```

Example #2 :

```
player {  
    name = "Steve"  
    level = 30  
}
```

In practice, it is best to follow the formatting conventions of the HOCON configuration you are editing. When editing a HOCON configuration for Sponge or an individual plugin utilizing the Sponge API, the values are likely the only thing you will be changing unless otherwise specified.

Debugging your configuration

If a HOCON configuration does not appear to be working, here are some tips.

- Curly braces must be balanced
- Quotation marks must be balanced
- Duplicate keys that appear later take precedence

Specification

More information about the HOCON format can be found [here](#).

JSON Syntax

This is an example of a `whitelist.json` file with correct formatting (although the UUID-s are fictional). Your file should follow the same syntax.

```
[  
  {  
    "uuid": "01234567-89ab-cdef-0123-456789abcdef",  
    "name": "Notch"  
  },  
  {  
    "uuid": "a0b1c2d3-e4f5-0617-2839-4a5b6c7d8e9f",  
    "name": "sk89q"  
  }  
]
```

Format Rules

- Square braces (`[]`) open and close the file
- Each entry in the file is wrapped with curly braces (`{ }`)
- Each key and its corresponding value is typed on its own line
- If more than one exists, both entries and key/value pairs are comma separated
- All strings are in quotation marks
- UUID-s are 32 symbols long, and written in hexadecimal (0-9, a-f).
- the UUID symbols are grouped. First is a group of 8, then three groups of 4, then a group of 12. The groups are separated by dashes (`-`)

What You Can Configure

global.conf

Global Configuration

The `global.conf` file contains the global configuration settings for Sponge. This file is created in the `config/sponge` directory in your server folder. Many of these properties can be also overridden per-world or per-dimension type by using the config files in the subfolders of `config/worlds`.

Below is a table with all available settings inside the `global.conf` file. Note that certain sections will not be filled immediately, and may optionally be added to the file when the server encounters them. There's also full example of a unmodified `global.conf` file at the bottom of this page, below the following table :

Global Properties of Sponge

Property	Type	Default	Description
Block Tracking			
block-blacklist	string	null	Adds block ids you wish to blacklist for player block placement tracking.
enabled	boolean	true	Adds player tracking support for block positions.
Bungeecord			
ip-forwarding	boolean	false	Allows bungeecord to forward ip address, UUID, and Game Profile to the server.
Cause Tracker			
report-different-world-changes	boolean	false	If enabled, Sponge will report when a mod makes an unexpected world change.
verbose	boolean	true	If enabled, the cause tracker will print out when there are too many phases being entered.
Commands			
aliases	string	null	Alias will resolve conflicts when multiple plugins request a specific command. Correct syntax is <code><unqualified command>=<plugin name></code> Example : <pre>aliases = { title=myPlugin }</pre>

Suite sur la page suivante

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
config-enabled	boolean	false	In dimension/world configs, it allows the config to override inherited configs.
Debug Options			
concurrent-entity-checks	boolean	false	Detects and prevents attempts to use entities concurrently.
dump-chunks-on-deadlock	boolean	false	Dumps chunks in the event of a deadlock.
dump-heap-on-deadlock	boolean	false	Dump the heap in the event of a deadlock.
dump-threads-on-warn	boolean	false	Dump the server thread on deadlock warning.
thread-contention-monitoring	boolean	false	Enable Java's thread contention monitoring for thread dumps.
Entity Options			
collision-warn-size	integer	200	Number of colliding entities in one spot before logging a warning. Set to 0 to disable.
count-warn-size	integer	0	Number of entities allowed in one dimension before logging a warning. Set to 0 to disable.
entity-painting-respawn-delay	integer	2	Number of ticks before a painting is respawned on the client when their art is changed.
human-player-list-remove-delay	integer	10	Number of ticks before the fake player entry of a human is removed from the tab list. The allowed range is 0 - 100.
item-despawn-rate	integer	6000	The time in ticks before an item despawns.
living-hard-despawn-range	integer	128	The upper bounded range where living entities farther from a player will likely despawn
living-soft-despawn-minimum-life	integer	30	The amount of seconds before a living entity between the soft and hard despawn ranges from a player to be considered for despawning
Suite sur la page suivante			

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
living-soft-despawn-range	integer	32	The lower bounded range where living entities near a player may potentially despawn
max-bounding-box-size	integer	1000	Maximum size of an entity's bounding box before it is removed. Set to 0 to disable.
max-speed	integer	100	Square of the maximum speed of an entity before it is removed. Set to 0 to disable
Entity Activation Range			
auto-populate	boolean	false	If enabled, newly discovered entities will be added to this config with a default value.
Defaults			Default activation ranges for all entities unless overridden.
ambient	integer	32	Default activation range for ambient entities.
aquatic	integer	32	Default activation range for aquatic entities.
creature	integer	16	Default activation range for creatures.
misc	integer	16	Default activation range for miscellaneous entities.
monster	integer	32	Default activation range for monsters.
Mods			Per-mod overrides. Refer to the Minecraft default mod for example.
Entity Collisions			
auto-populate	boolean	false	If enabled, newly discovered entities/blocks will be added to this config with a default value.
Defaults			Default max collisions used for all entities/blocks unless overridden.
max-entities-within-aabb	integer	8	Max amount of entities any given entity or block can collide with. Set to 0 to disable.
Mods			Per-mod overrides. Refer to the Minecraft default mod for example.
Minecraft			
Blocks			

Suite sur la page suivante

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
“detector_rail”	integer	1	Max collisions for a “detector_rail”.
“heavy_weighted_pressure_plate”	integer	150	Max collisions for a “heavy_weighted_pressure_plate”.
“light_weighted_pressure_plate”	integer	15	Max collisions for a “light_weighted_pressure_plate”.
“mob_spawner”	integer	-1	Max collisions for a “mob_spawner”.
“stone_pressure_plate”	integer	1	Max collisions for a “stone_pressure_plate”.
“wooden_button”	integer	1	Max collisions for a “wooden_button”.
“wooden_pressure_plate”	integer	1	Max collisions for a “wooden_pressure_plate”.
Defaults			Default max collisions used for all entities/blocks unless overridden.
blocks	integer	8	Default max collisions for blocks.
entities	integer	8	Default max collisions for entities.
enabled	boolean	true	Set to false if you want mod to ignore entity collision rules.
Entities			
thrownpotion	integer	-1	Max collisions for a thrown potion.
Exploits			
prevent-creative-itemstack-name-exploit	boolean	true	Prevents an exploit in which the client sends a packet with the itemstack name exceeding the string limit.
prevent-sign-command-exploit	boolean	true	Prevents an exploit in which the client sends a packet to update a sign containing commands from a player without permission.
General Settings			
config-dir	string	see config	Sets the directory for plugin configurations.
disable-warnings	boolean	false	Disable warning messages to server Admins.
file-io-thread-sleep	boolean	false	Enables sleeping between chunk saves.
plugins-dir	string	See config	Sets an additional directory to search for plugins.
Ip Sets			

Suite sur la page suivante

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
Logging Options			
block-break	boolean	false	Logs when a block is broken.
block-modify	boolean	false	Logs when blocks are modified.
block-place	boolean	false	Logs when blocks are placed.
block-populate	boolean	false	Logs when blocks are populated in a chunk.
block-tracking	boolean	false	Logs when blocks are placed by players and tracked.
chunk-gc-queue-unload	boolean	false	Logs when chunks are queued to be unloaded.
chunk-load	boolean	false	Log when chunks are loaded.
chunk-unload	boolean	false	Log when chunks are unloaded.
entity-collision-checks	boolean	false	Whether to log entity collision/count checks.
entity-death	boolean	false	Log when living entities are destroyed.
entity-despawn	boolean	false	Log when living entities are despawned.
entity-spawn	boolean	false	Log when living entities are spawned.
entity-speed-removal	boolean	false	Whether to log entity removals due to speed.
exploit-itemstack-name-overflow	boolean	false	Logs when a server receives exploited packets with itemstack name exceeding string limit.
exploit-respawn-invisibility	boolean	false	Logs when a player attempts to respawn invisible to surrounding players.
exploit-sign-command-updates	boolean	false	Logs when a server receives an exploited packet containing commands from a player with no permission.
log-stacktraces	boolean	false	Add stack traces to dev logging.
world-auto-save	boolean	false	If true, logs when a world auto-saves its chunk data.
Modules			
bungeecord	boolean	false	Enables bungeecord support.

Suite sur la page suivante

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
entity-activation-range	boolean	true	Enables the entity activation range settings.
entity-collisions	boolean	true	Enables entity collision settings.
exploits	boolean	true	Enables the exploit prevention module.
game-fixes	boolean	false	Enables the game fixes module.
optimizations	boolean	true	Enables the optimizations module.
realtime	boolean	false	Use real time instead of ticks.
timings	boolean	true	Enables timing settings.
tracking	boolean	true	Enables the tracking module.
Optimizations			See ../management/performance-tweaks
Spawner			
spawn-limit-ambient	int	15	The number of ambients that can spawn around the player.
spawn-limit-animal	int	15	The number of animals that can spawn around the player.
spawn-limit-aquatic	int	5	The number of aquatics that can spawn around the player.
spawn-limit-monster	int	70	The number of monsters that can spawn around the player.
tick-rate-ambient	int	400	The ambient spawn tick rate.
tick-rate-animal	int	400	The animal spawn tick rate.
tick-rate-aquatic	int	400	The aquatic spawn tick rate.
tick-rate-monster	int	1	The monster spawn tick rate.
SQL			
aliases	string	null	Aliases for SQL connections. This is done in the format jdbc:protocol://[username[:password]@]host/database
Timings			
enabled	boolean	true	If timings are enabled.
hidden-config-entries	string	sponge.sql	The hidden config entries.

Suite sur la page suivante

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
history-interval	integer	300	The interval between timing history report generation.
history-length	integer	3600	How long, in ticks, that the timing history will be kept for the server.
server-name-privacy	boolean	false	Whether to include information such as the server name, motd, online-mode, and server icon in the report.
verbose	boolean	true	Whether or not for timings to monitor at the verbose level.
World Settings			
auto-player-save-interval	integer	900	The auto-save tick interval used when saving global player data.
auto-save-interval	integer	900	The auto-save tick interval used to save all loaded chunks in a world.
chunk-gc-load-threshold	integer	0	The number of newly loaded chunks before triggering a forced cleanup.
chunk-gc-tick-interval	integer	1	The tick interval used to cleanup all inactive chunks in a world.
chunk-unload-delay	integer	30	The number of seconds to delay a chunk unload once marked inactive.
deny-chunk-requests	boolean	true	If enabled, any request for a chunk not currently loaded will be denied.
flowing-lava-decay	boolean	false	Lava behaves like vanilla water when the source block is removed, when set to true.
gameprofile-lookup-batch-size	integer	1	The amount of GameProfile requests to make against Mojang's session server.
gameprofile-lookup-task-interval	integer	1	The interval used to process queued GameProfile requests.
generate-spawn-on-load	boolean	false	If the world should generate spawn when the world is loaded.
infinite-water-source	boolean	false	False = Default vanilla water source behaviour.
Suite sur la page suivante			

Tableau 2.1 – Suite de la page précédente

Property	Type	Default	Description
invalid-lookup-uuids	array	See config	The list of uuid's that shouldn't be looked up on Mojang's session server.
item-merge-radius	integer	2.5	The merge radius for item entities.
keep-spawn-loaded	boolean	false	If the spawn should stay loaded with no players. Has no effect in global config. Config doesn't need to be enabled either, because it will always fall back to the world config.
leaf-decay	boolean	true	If enabled, allows natural leaf decay.
load-on-startup	boolean	false	If the world should be loaded on startup. Has no effect in global config. Config doesn't need to be enabled either, because it will always fall back to the world config.
mob-spawn-range	integer	8	Specifies the radius (in chunks) of where creatures will spawn. This value is capped to the current view distance setting in server.properties.
Portal Agents			A list of all detected portal agents used in this world. In order to override, change the target world name to any other valid world. If world is not found, it will fallback to default.
"minecraft :default_nether"	world	DIM-1	The default nether world.
"minecraft :default_the_end"	world	DIM1	The default end world.
pvp-enabled	boolean	true	If the world allows PVP combat.
weather-ice-and-snow	boolean	true	Enable to allow the natural formation of ice and snow.
weather-thunder	boolean	true	Enable to initiate thunderstorms.
world-enabled	boolean	true	Enable if this world should be registered.

This config was generated using SpongeForge build 2360 (with Forge 2282), SpongeAPI version 6.0 :

```

# 1.0
#
# # If you need help with the configuration or have any questions related to Sponge,
# # join us at the IRC or drop by our forums and leave a post.
#
# # IRC: #sponge @ irc.esper.net ( http://webchat.esper.net/?channel=sponge )
# # Forums: https://forums.spongepowered.org/
#

sponge {
    block-capturing {
        # If enabled, newly discovered blocks will be added to this config with a
        ↪default value.
        auto-populate=false
        # Per-mod block id mappings for controlling capturing behavior
        mods {
            extratils2 {
                # Set to true to perform individual capturing (i.e. skip bulk
                ↪capturing) for scheduled ticks for a block type
                block-tick-capturing {
                    RedstoneClock=true
                }
                # Set to false if you want to ignore all specific rules for this mod
                enabled=true
            }
        }
    }
    block-tracking {
        # Add block ids you wish to blacklist for player block placement tracking.
        block-blacklist=[]
        # If enabled, adds player tracking support for block positions. Note: This
        ↪should only be disabled if you do not care who caused a block to change.
        enabled=true
    }
    bungeecord {
        # If enabled, allows BungeeCord to forward IP address, UUID, and Game Profile
        ↪to this server
        ip-forwarding=false
    }
    cause-tracker {
        # If true, when a mod changes a world that is different
        # from an expected world during a WorldTick event, the
        # cause tracker will identify both the expected changed
        # world and the actual changed world. This does not mean
        # that the changes are being dropped, simply it means that
        # a mod is possibly unknowingly changing a world other
        # than what is expected.
        report-different-world-changes=false
        # If true, the cause tracker will print out when there are too many phases
        # being entered, usually considered as an issue of phase re-entrance and
        # indicates an unexpected issue of tracking phases not to complete.
        # If this is not reported yet, please report to Sponge. If it has been
        # reported, you may disable this.
        verbose=true
        # If true, the cause tracker will dump extra information about the current
        ↪phaseswhen certain non-CauseTracker related exceptions occur. This is usually not
        ↪necessary, as the information in the exception itself can normally be used to
        ↪determine the cause of the issue
    }
}

```

```

        verbose-errors=false
    }
    commands {
        # A mapping from unqualified command alias to plugin id of the plugin that
        ↪should handle a certain command
        aliases {}
        # Patches the specified commands to respect the world of the sender instead
        ↪of applying the changes on the all worlds.
        multi-world-patches {
            defaultgamemode=true
            difficulty=true
            gamerule=true
            seed=true
            setdefaultspawnpoint=true
            time=true
            toggledownfall=true
            weather=true
            worldborder=true
        }
    }
    # This setting does nothing in the global config. In dimension/world configs, it
    ↪allows the config to override config(s) that it inherits from
    config-enabled=false
    debug {
        # Detect and prevent certain attempts to use entities concurrently.
        # WARNING: May drastically decrease server performance. Only enable this to
        ↪debug a pre-existing issue
        concurrent-entity-checks=false
        # Dump chunks in the event of a deadlock
        dump-chunks-on-deadlock=false
        # Dump the heap in the event of a deadlock
        dump-heap-on-deadlock=false
        # Dump the server thread on deadlock warning
        dump-threads-on-warn=false
        # Enable Java's thread contention monitoring for thread dumps
        thread-contention-monitoring=false
    }
    entity {
        # Number of colliding entities in one spot before logging a warning. Set to 0
        ↪to disable
        collision-warn-size=200
        # Number of entities in one dimension before logging a warning. Set to 0 to
        ↪disable
        count-warn-size=0
        # Number of ticks before a painting is respawned on clients when their art is
        ↪changed
        entity-painting-respawn-delay=2
        # Number of ticks before the fake player entry of a human is removed from the
        ↪tab list (range of 0 to 100 ticks).
        human-player-list-remove-delay=10
        # Controls the time in ticks for when an item despawns.
        item-despawn-rate=6000
        # The upper bounded range where living entities farther from a player will
        ↪likely despawn
        living-hard-despawn-range=128
        # The amount of seconds before a living entity between the soft and hard
        ↪despawn ranges from a player to be considered for despawning
        living-soft-despawn-minimum-life=30
    }

```

```

# The lower bounded range where living entities near a player may potentially
↳despawn
living-soft-despawn-range=32
# Max size of an entity's bounding box before removing it. Set to 0 to disable
max-bounding-box-size=1000
# Square of the max speed of an entity before removing it. Set to 0 to disable
max-speed=100
}
entity-activation-range {
# If enabled, newly discovered entities will be added to this config with a
↳default value.
auto-populate=false
# Default activation ranges used for all entities unless overridden.
defaults {
    ambient=32
    aquatic=32
    creature=32
    misc=16
    monster=32
}
# Per-mod overrides. Refer to the minecraft default mod for example.
mods {}
}
entity-collisions {
# If enabled, newly discovered entities/blocks will be added to this config
↳with a default value.
auto-populate=false
# Default max collisions used for all entities/blocks unless overridden.
defaults {
    blocks=8
    entities=8
}
# Max amount of entities any given entity or block can collide with. This
↳improves performance when there are more than 8 entities on top of eachother such
↳as a 1x1 spawn pen. Set to 0 to disable.
max-entities-within-aabb=8
# Per-mod overrides. Refer to the minecraft default mod for example.
mods {
    botania {
        blocks {}
        # Default max collisions used for all entities/blocks unless
↳overridden.
        defaults {}
        # Set to false if you want mod to ignore entity collision rules.
        enabled=true
        entities {
            botaniacorporeaspark=-1
            botaniaspark=-1
        }
    }
    minecraft {
        blocks {
            "detector_rail"=1
            "heavy_weighted_pressure_plate"=150
            "light_weighted_pressure_plate"=15
            "mob_spawner"=-1
            "stone_pressure_plate"=1
            "wooden_button"=1

```

```

        "wooden_pressure_plate"=1
    }
    # Default max collisions used for all entities/blocks unless_
→ overridden.
    defaults {}
    # Set to false if you want mod to ignore entity collision rules.
    enabled=true
    entities {
        thrownpotion=-1
    }
}
}
exploits {
    prevent-creative-itemstack-name-exploit=true
    prevent-sign-command-exploit=true
}
general {
    # The directory for Sponge plugin configurations, relative to the
    # execution root or specified as an absolute path.
    # Note that the default: "${CANONICAL_GAME_DIR}/config"
    # is going to use the "plugins" directory in the root game directory.
    # If you wish for plugin configs to reside within a child of the configuration
    # directory, change the value to, for example, "${CANONICAL_CONFIG_DIR}/
→ sponge/plugins".
    # Note: It is not recommended to set this to "${CANONICAL_CONFIG_DIR}/sponge",
→ as there is
    # a possibility that plugin configurations can conflict the Sponge core_
→ configurations.
    config-dir="${CANONICAL_GAME_DIR}/config"
    # Disable warning messages to server admins
    disable-warnings=false
    # Enabled sleeping between chunk saves, beware of memory issues
    file-io-thread-sleep=false
    # Additional directory to search for plugins, relative to the
    # execution root or specified as an absolute path.
    # Note that the default: "${CANONICAL_MODS_DIR}/plugins"
    # is going to search for a plugins folder in the mods directory.
    # If you wish for the plugins folder to reside in the root game
    # directory, change the value to "${CANONICAL_GAME_DIR}/plugins".
    plugins-dir="${CANONICAL_MODS_DIR}/plugins"
}
ip-sets {}
logging {
    # Log when blocks are broken
    block-break=false
    # Log when blocks are modified
    block-modify=false
    # Log when blocks are placed
    block-place=false
    # Log when blocks are populated in a chunk
    block-populate=false
    # Log when blocks are placed by players and tracked
    block-tracking=false
    # Log when chunks are queued to be unloaded by the chunk garbage collector.
    chunk-gc-queue-unload=false
    # Log when chunks are loaded
    chunk-load=false

```



```

# Log when chunks are unloaded
chunk-unload=false
# Whether to log entity collision/count checks
entity-collision-checks=false
# Log when living entities are destroyed
entity-death=false
# Log when living entities are despawned
entity-despawn=false
# Log when living entities are spawned
entity-spawn=false
# Whether to log entity removals due to speed
entity-speed-removal=false
# Log when server receives exploited packet with itemstack name exceeding_
↳string limit.
  exploit-itemstack-name-overflow=false
# Log when player attempts to respawn invisible to surrounding players.
  exploit-respawn-invisibility=false
# Log when server receives exploited packet to update a sign containing_
↳commands from player with no permission.
  exploit-sign-command-updates=false
# Add stack traces to dev logging
  log-stacktraces=false
# Log when a world auto-saves its chunk data. Note: This may be spammy_
↳depending on the auto-save-interval configured for world.
  world-auto-save=false
}
modules {
  block-capturing-control=true
  bungeecord=false
  entity-activation-range=true
  entity-collisions=true
  exploits=true
  game-fixes=false
  optimizations=true
# Use real (wall) time instead of ticks as much as possible
  realtime=false
# Controls block range and tick rate of tileentities.
# Use with caution as this can break intended functionality.
  tileentity-activation=false
  timings=true
  tracking=true
}
optimizations {
# Runs lighting updates async.
  async-lighting=true
# Caches tameable entities owners to avoid constant lookups against data_
↳watchers. If mods cause issue, disable.
  cache-tameable-owners=true
# If enabled, block item drops are pre-processed to avoid
# having to spawn extra entities that will be merged post spawning.
# Usually, Sponge is smart enough to determine when to attempt an item pre-
↳merge
# and when not to, however, in certain cases, some mods rely on items not_
↳being
# pre-merged and actually spawned, in which case, the items will flow right_
↳through
# without being merged.
  drops-pre-merge=true

```

```

# Handles structures that are saved to disk. Certain structures can take up
↳large amounts
# of disk space for very large maps and the data for these structures is only
↳needed while the world
# around them is generating. Disabling saving of these structures can save
↳disk space and time during
# saves if your world is already fully generated.
# Warning: disabling structure saving will break the vanilla locate command.
structure-saving {
    # If enabled, newly discovered structures will be added to this config
↳with a default value.
    auto-populate=false
    enabled=false
    # Per-mod overrides. Refer to the minecraft default mod for example.
    mods {
        minecraft {
            # Set to false if you want mod to never save structures.
            enabled=true
            structures {
                mineshaft=false
            }
        }
    }
}

# Used to control spawn limits around players.
# Note: The radius uses the lower value of mob spawn range and server's view
↳distance.
spawner {
    # The number of ambients the spawner can potentially spawn around a player.
    spawn-limit-ambient=15
    # The number of animals the spawner can potentially spawn around a player.
    spawn-limit-animal=15
    # The number of aquatics the spawner can potentially spawn around a player.
    spawn-limit-aquatic=5
    # The number of monsters the spawner can potentially spawn around a player.
    spawn-limit-monster=70
    # The ambient spawning tick rate. Default: 400
    tick-rate-ambient=400
    # The animal spawning tick rate. Default: 400
    tick-rate-animal=400
    # The aquatic spawning tick rate. Default: 400
    tick-rate-aquatic=400
    # The monster spawning tick rate. Default: 1
    tick-rate-monster=1
}

# Configuration options related to the Sql service, including connection aliases
↳etc
sql {
    # Aliases for SQL connections, in the format jdbc:protocol://
↳[username[:password]@]host/database
    aliases {}
}

tileentity-activation {
    # If enabled, newly discovered tileentities will be added to this config with
↳default settings.
    auto-populate=false
    # Default activation block range used for all tileentities unless overridden.

```

```

default-block-range=64
# Default tick rate used for all tileentities unless overridden.
default-tick-rate=1
# Per-mod overrides. Refer to the minecraft default mod for example.
mods {}
}
timings {
    enabled=true
    hidden-config-entries=[
        "sponge.sql"
    ]
    history-interval=300
    history-length=3600
    server-name-privacy=false
    verbose=true
}
world {
    # The auto-save tick interval used when saving global player data. (Default: 900)
    # Note: 20 ticks is equivalent to 1 second. Set to 0 to disable.
    auto-player-save-interval=900
    # The auto-save tick interval used to save all loaded chunks in a world.
    # Set to 0 to disable. (Default: 900)
    # Note: 20 ticks is equivalent to 1 second.
    auto-save-interval=900
    # The number of newly loaded chunks before triggering a forced cleanup.
    # Note: When triggered, the loaded chunk threshold will reset and start incrementing.
    # Disabled by default.
    chunk-gc-load-threshold=0
    # The tick interval used to cleanup all inactive chunks that have leaked in a world.
    # Set to 0 to disable which restores vanilla handling. (Default: 600)
    chunk-gc-tick-interval=600
    # The number of seconds to delay a chunk unload once marked inactive. (Default: 15)
    # Note: This gets reset if the chunk becomes active again.
    chunk-unload-delay=15
    # If enabled, any request for a chunk not currently loaded will be denied (exceptions apply for things like world gen and player movement).
    # Note: As this is an experimental setting for performance gain, if you encounter any issues then we recommend disabling it.
    deny-chunk-requests=true
    # Lava behaves like vanilla water when source block is removed
    flowing-lava-decay=false
    # The amount of GameProfile requests to make against Mojang's session server. (Default: 1)
    # Note: Mojang accepts a maximum of 600 requests every 10 minutes from a single IP address.
    # If you are running multiple servers behind the same IP, it is recommended to raise the 'gameprofile-task-interval' setting
    # in order to compensate for the amount requests being sent.
    # Finally, if set to 0 or less, the default batch size will be used.
    # For more information visit http://wiki.vg/Mojang\_API
    gameprofile-lookup-batch-size=1
    # The interval, in seconds, used by the GameProfileQueryTask to process queued gameprofile requests. (Default: 4)
    # Note: This setting should be raised if you experience the following error:

```

```

# "The client has sent too many requests within a certain amount of time".
# Finally, if set to 0 or less, the default interval will be used.
gameprofile-lookup-task-interval=4
# Enable if you want the world to generate spawn the moment its loaded.
generate-spawn-on-load=false
# Vanilla water source behavior - is infinite
infinite-water-source=false
# The list of uuid's that should never perform a lookup against Mojang's
↪session server.
# Note: If you are using SpongeForge, make sure to enter any mod fake player
↪'s UUID to this list.
invalid-lookup-uuids=[
    "00000000-0000-0000-0000-000000000000",
    "0d0c4ca0-4ff1-11e4-916c-0800200c9a66",
    "41c82c87-7afb-4024-ba57-13d2c99cae77"
]
# The defined merge radius for Item entities such that when two items are
# within the defined radius of each other, they will attempt to merge.
↪Usually,
# the default radius is set to 0.5 in Vanilla, however, for performance
↪reasons
# 2.5 is generally acceptable.
# Note: Increasing the radius higher will likely cause performance degradation
# with larger amount of items as they attempt to merge and search nearby
# areas for more items. Setting to a negative value is not supported!
item-merge-radius=2.5
# Enable if this world's spawn should remain loaded with no players.
keep-spawn-loaded=false
# Enable to allow natural leaf decay.
leaf-decay=true
# Enable if this world should be loaded on startup.
load-on-startup=false
# The maximum number of queued unloaded chunks that will be unloaded in a
↪single tick.
# Note: With the chunk gc enabled, this setting only applies to the ticks
# where the gc runs (controlled by 'chunk-gc-tick-interval')
# Note: If the max unloads is too low, too many chunks may remain
# loaded on the world and increases the chance for a drop in tps. (Default:
↪100)
max-chunk-unloads-per-tick=100
# Specifies the radius (in chunks) of where creatures will spawn.
# This value is capped to the current view distance setting in server.
↪properties
mob-spawn-range=4
# A list of all detected portal agents used in this world.
# In order to override, change the target world name to any other valid world.
# Note: If world is not found, it will fallback to default.
portal-agents {
    "minecraft:default_nether"=DIM-1
    "minecraft:default_the_end"=DIM1
}
# Enable if this world allows PVP combat.
pvp-enabled=true
# Enable to allow the natural formation of ice and snow in supported biomes.
weather-ice-and-snow=true
# Enable to initiate thunderstorms in supported biomes.
weather-thunder=true
# Enable if this world should be registered.

```

```
    world-enabled=true
  }
}
```

server.properties

Default File

Here is the default server.properties file of an unmodified Minecraft 1.8.1 server.

```
#Minecraft server properties
#(File modification datestamp)
spawn-protection=16
max-tick-time=60000
generator-settings=
force-gamemode=false
allow-nether=true
gamemode=0
enable-query=false
player-idle-timeout=0
difficulty=1
spawn-monsters=true
op-permission-level=4
resource-pack-hash=
announce-player-achievements=true
pvp=true
snooper-enabled=true
level-type=DEFAULT
hardcore=false
enable-command-block=false
max-players=20
network-compression-threshold=256
max-world-size=29999984
server-port=25565
server-ip=
spawn-npcs=true
allow-flight=false
level-name=world
view-distance=10
resource-pack=
spawn-animals=true
white-list=false
generate-structures=true
online-mode=true
max-build-height=256
level-seed=
use-native-transport=true
motd=A Minecraft Server
enable-rcon=false
```

Property Explanation

Credit goes to the editors at the [Minecraft Wiki](#) for the explanations.

Boolean properties have only two valid values : *true* and *false*. **Integer** properties must be whole numbers. Where a range is specified, the number must be in that range. **String** properties can consist of any symbol.

Key	Type	Default Value	Description
allow-flight	boolean	false	<p>Allows users to use flight on your server while in Survival mode, if they have a mod that provides flight installed.</p> <p>With allow-flight enabled griefers will possibly be more common, because it will make their work easier. In Creative mode this has no effect.</p> <p>false - Flight is not allowed (players in air for at least 5 seconds will be kicked).</p> <p>true - Flight is allowed.</p>
allow-nether	boolean	true	<p>Allows players to travel to the Nether.</p> <p>false - Nether portals will not work.</p> <p>true - The server will allow portals to send players to the Nether.</p>
announce-player-achievements	boolean	true	<p>Allows the server to announce when a player gets an achievement.</p>
difficulty	integer (0-3)	1	<p>Defines the difficulty (such as damage dealt by mobs and the way hunger and poison affects players) of the server.</p> <p>0 - Peaceful</p> <p>1 - Easy</p> <p>2 - Normal</p> <p>3 - Hard</p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
enable-query	boolean	false	Enables the GameSpy4 protocol server listener. Used to get information about the server.
enable-rcon	boolean	false	Enables remote access to the server console.
enable-command-block	boolean	false	Enables command blocks.
force-gamemode	boolean	false	Force players to join in the default game mode. false - Players will join in the gamemode they had when they last left. true - Players will always join in the default gamemode.
gamemode	integer (0-3)	0	Defines the mode of gameplay. 0 - Survival 1 - Creative 2 - Adventure 3 - Spectator
generate-structures	boolean	true	Defines whether structures (such as villages) will be generated in new chunks. false - Structures will not be generated. true - Structures will be generated. Note : Dungeons will still generate if this is set to false.
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
generator-settings	string	<i>blank</i>	The settings used to customize world generation. See Superflat and Customized on the Minecraft Wiki (external links) for possible settings and examples.
hardcore	boolean	false	If set to true , players will be permanently banned if they die.
level-name	string	world	The “level-name” value will be used as the world name and its folder name. You may also copy your saved game folder here, and change the name to the same as that folder’s to load it instead. Characters such as ‘ (apostrophe) may need to be escaped by adding a backslash (\) before them.
level-seed	string	<i>blank</i>	Add a seed for your world, as in Singleplayer. Some examples are : mine-craft, 404, 1a2b3c.
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
level-type	string	DEFAULT	<p>Determines the type of map that is generated.</p> <p>DEFAULT</p> <ul style="list-style-type: none"> - Standard world with hills, valleys, water, etc. <p>FLAT - A flat world with no features, meant for building.</p> <p>LARGEBIOMES</p> <ul style="list-style-type: none"> - Same as default, but all biomes are larger. <p>AMPLIFIED</p> <ul style="list-style-type: none"> - Same as default, but world-generation height limit is increased. <p>CUSTOMIZED</p> <ul style="list-style-type: none"> - Same as default unless generator-settings is set to a preset.
max-build-height	integer	256	The maximum height in which building is allowed. Terrain may still naturally generate above a low height limit.
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
max-players	integer (0- 2147483647)	20	The maximum number of players that can play on the server at the same time. Note that if more players are on the server it will use more resources. Note also, op player connections are not supposed to count against the max players, but ops currently cannot join a full server. Extremely large values for this field result in the client-side user list being broken.
max-tick-time	integer (0- (2^63-1))	60000	The maximum number of milliseconds a single tick may take before the server watchdog stops the server with the message : “A single server tick took 60.00 seconds (should be max 0.05) ; Considering it to be crashed, server will forcibly shutdown”. Once this criteria is met, it calls System.exit(1). -1 - disable watchdog entirely

Suite sur la page suivante

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
max-world-size	integer (1- 29999984)	29999984	<p>This sets the maximum possible size in blocks, expressed as a radius, that the world border can obtain. Setting the world border bigger causes the commands to complete successfully, but the actual border will not move past this block limit. Setting the max-world-size higher than the default doesn't appear to do anything.</p> <p>Examples :</p> <ul style="list-style-type: none"> — Setting max-world-size to 1000 will allow you to have a 2000x2000 world border. — Setting max-world-size to 4000 will give you an 8000x8000 world border.
motd	string	<i>A Minecraft Server</i>	<p>This is the message that is displayed in the server list of the client, below the name.</p> <ul style="list-style-type: none"> — The MOTD does support color and formatting codes. — If the MOTD is over 59 characters, the server list will likely report a communication error.
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
network-compression-threshold	integer	256	<p>By default it allows packets that are n-1 bytes big to go normally, but a packet that n bytes or more will be compressed down. So, lower number means more compression but compressing small amounts of bytes might actually end up with a larger result than what went in.</p> <p>-1 - disable compression entirely</p> <p>0 - compress everything</p> <p>Note : <i>The ethernet spec requires that packets less than 64 bytes become padded to 64 bytes. Thus, setting a value lower than 64 may not be beneficial. It is also not recommended to exceed the MTU (Maximum Transmission Unit), typically 1500 bytes.</i></p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
online-mode	boolean	true	<p>Server checks connecting players against Minecraft's account database. Only set this to false if your server is not connected to the Internet. Hackers with fake accounts can connect if this is set to false! If minecraft.net is down or inaccessible, no players will be able to connect if this is set to true. Setting this variable to off purposely is called "cracking" a server, and servers that are presently in offline mode are called "cracked" servers, allowing players with unlicensed copies of Minecraft to join.</p> <p>false - Disabled. The server will not attempt to check connecting players.</p> <p>true - Enabled. The server will assume it has an Internet connection and check every connecting player.</p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
op-permission-level	integer (1-4)	4	<p>Sets permission level for ops. Each level also contains the permissions of the levels below it.</p> <p>1 - Ops can bypass spawn protection.</p> <p>2 - Ops can use /clear, /difficulty, /effect, /gamemode, /gamerule, /give, and /tp, and can edit command blocks.</p> <p>3 - Ops can use /ban, /deop, /kick, and /op.</p> <p>4 - Ops can use /stop.</p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
player-idle-timeout	integer	0	<p>If non-zero, players are kicked from the server if they are idle for more than that many minutes.</p> <p>Note : <i>Idle time is reset when the server receives one of the following packets :</i></p> <ul style="list-style-type: none"> — 102 (0x66) Window-Click — 108 (0x6c) Button-Click — 130 (0x82) Update-Sign — 14 (0xe) BlockDig — 15 (0xf) Place — 16 (0x10) BlockItem-Switch — 18 (0x12) ArmAnimation — 19 (0x13) EntityAction — 205 (0xcd) Client-Command — 3 (0x3) Chat — 7 (0x7) UseEntity
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
pvp	boolean	true	<p>Enable PvP on the server. Players shooting themselves with arrows will only receive damage if PvP is enabled.</p> <p>false - Players cannot kill other players (also known as Player versus Environment (PvE)).</p> <p>true - Players will be able to kill each other.</p> <p>Note : <i>Indirect damage sources spawned by players (such as lava, fire, TNT and to some extent water, sand and gravel) will still deal damage to other players.</i></p>
query.port	integer (1- 65534)	25565	Sets the port for the query server (see enable-query).
rcon.password	string	<i>blank</i>	Sets the password for remote connection.
rcon.port	integer (1- 65534)	25575	Sets the port for remote connection.
resource-pack	string	<i>blank</i>	Optional URI to a resource pack. The player may choose to use it.
resource-pack-hash	string	<i>blank</i>	Optional SHA-1 digest of the resource pack, in lowercase hexadecimal. It's recommended to specify this. This is not yet used to verify the integrity of the resource pack, but improves the effectiveness and reliability of caching.
server-ip	string	<i>blank</i>	Set this if you want the server to bind to a particular IP. It is strongly recommended that you leave this blank.
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
server-port	integer (1- 65534)	25565	Changes the port the server is hosting (listening) on. This port must be forwarded if the server is hosted in a network using NAT (If you have a home router/firewall).
snooper-enabled	boolean	true	Sets whether the server sends snoop data regularly to http://snoop.minecraft.net . (external link) false - Disable sending of data. true - Enable sending of data.
spawn-animals	boolean	true	Determines whether animals will be able to spawn. false - All animals will immediately vanish, and none will spawn. true - Animals spawn as normal. <i>Tip : if you have major lag, set this to false.</i>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
spawn-monsters	boolean	true	<p>Determines whether hostile mobs will be able to spawn.</p> <p>false - All mobs will immediately vanish, and none will spawn anywhere, or at any time of day.</p> <p>true - Mobs spawn as normal ; in darkness and at night.</p> <p>This will have no effect if difficulty is set to Peaceful.</p> <p><i>Tip : if you have major lag, set this to false.</i></p>
spawn-npcs	boolean	true	<p>Determines if villagers will be spawned.</p> <p>true - Enabled. Villagers will spawn.</p> <p>false - Disabled. No villagers.</p>
spawn-protection	integer	16	<p>Determines the radius of the spawn protection. Setting this to 0 will not disable spawn protection. 0 will protect the single block at the spawn point. 1 will protect a 3x3 area centered on the spawn point. 2 will protect 5x5, 3 will protect 7x7, etc. This option is not generated on the first server start and appears when the first player joins. If there are no ops set on the server, the spawn protection will be disabled automatically.</p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
use-native-transport	boolean	true	<p>Linux server performance improvements : optimized packet sending/receiving on Linux.</p> <p>false - Disabled. Disable Linux packet sending/receiving optimization.</p> <p>true - Enabled. Enable Linux packet sending/receiving optimization.</p>
view-distance	integer (3-15)	10	<p>Sets the amount of world data the server sends the client, measured in chunks in each direction of the player (radius, not diameter). It determines the server-side viewing distance.</p> <p><i>10 is the default/recommended. If you have major lag, reduce this value.</i></p>
Suite sur la page suivante			

Tableau 2.2 – Suite de la page précédente

Key	Type	Default Value	Description
white-list	boolean	false	<p>Enables a whitelist on the server. See <code>../server/management/whitelist</code>. With a whitelist enabled, users not on the whitelist will be unable to connect. Intended for private servers, such as those for real-life friends or strangers carefully selected via an application process, for example.</p> <p>false - No whitelist is used.</p> <p>true - The file <code>whitelist.json</code> is used to generate the whitelist.</p> <p>Note : <i>Ops are automatically whitelisted, and there is no need to add them to the whitelist.</i></p>

Plugins will also have their own configuration files in the “config” folder.

World Configs

There are three types of world configs :

- Global
- Dimension
- World

Global configuration files can affect all of a server’s worlds and dimensions. This is the default level for configs. Dimension configuration files are used to affect a certain dimension or group of worlds. These types of configs will override the global config files. World configuration files are used to modify individual worlds only. World configs override dimension and global configs.

Modifying the Config In-Game

It is possible to modify these configs through the in-game command `/sponge config`. The syntax for the config command looks like this :

```
/sponge config <flag> <key> <value>
```

There are *flags* for specifying the target that you would wish to change. These flags are global, dimension, and world.

- `-g` is the flag for global
- `-d <dim>` targets a dimension (replacing `<dim>` with the dimension you want to configure)

— `-w <world>` targets one world (replacing `<world>` with your chosen world).
The *key* is the value you want to change. The *value* is whatever you want to change the value of the key to.

Here is an example of this command in action :

```
/sponge config -d nether logging.chunk-load true
```

This will set the config to log when chunks are loaded for the nether.

If you need to check the value of a key, you would need to omit the *value*. Checking the value of a key such as `logging.chunk-load` in the nether would be done like so :

```
/sponge config -d nether logging.chunk-load
```

Saving a World Config

Saving a world config to the file may be desired after making modifications. This would be useful in the event of an unexpected server crash. This would be done by using the `/sponge save` command on the sponge server. The syntax for this command is similar to the config command :

```
/sponge save <flag>
```

Here is an example for saving the global config :

```
/sponge save -g
```

Reloading a World Config

Sometimes it may be desired that a world config is reloaded while the server is still running. This would be useful if you have made changes to the local config file and would like to reload it for use on the live server. This is made possible by the command `/sponge reload`. The syntax for the command is as follows :

```
/sponge reload <flag>
```

Here is an example of reloading the end world config file :

```
/sponge reload -d the_end
```

Server Management

Contents

Managing the Whitelist

The whitelist allows you to control who can join your server. Be aware that ops will *always* be able to connect to the server, regardless of whether they're in the whitelist.

When the whitelist function is enabled, only players named on the whitelist will be allowed to login to your server. Players can be added to the whitelist through the usage of in-game commands or by editing the `whitelist.json` file. Beware, however : if you manually change the file, you will have to reload the whitelist or restart the server for the changes to go into effect. Additionally, pay special heed to the syntax, as the whitelist won't work if it is wrong. An example of a correctly formatted whitelist file can be found at `../server/getting-started/configuration/json`.

- To enable the whitelist, use `/whitelist on`
- To disable the whitelist, use `/whitelist off`
- To add a player to the whitelist, use `/whitelist add playername`
- To remove a player from the whitelist, use `/whitelist remove playername`
- To show all players on the whitelist, use `/whitelist list`
- To reload the whitelist after a manual change to the file, use `/whitelist reload`

The whitelist can also be enabled or disabled by editing the `../server/getting-started/configuration/server-properties` file, although this will only affect the game after server reload or restart.

Managing Bans

Minecraft, and consequently Sponge, has simple ban management to prevent unwanted users from joining your server.

The `/ban <name> [reason]` command is a native Minecraft server function that bans player *name*. The complete list of banned players is available using the command `/banlist players`

It is also possible to ban any connections from a given IP address using `/banip <address|name> [reason]`. The complete list of banned IP addresses is available using the command `/banlist ips`

A ban can be reversed using the command `/pardon <name>` or `/pardon <ip-address>`

More information on Bans can be found at the [Minecraft Wiki](#).

Managing Permissions

You can configure who has access to what if you are running a server by making use of permissions. Specific permissions for Sponge, Forge and Minecraft commands are shown on the `../server/spongineer/commands` page.

Operator Level

Minecraft comes with a simple way to give permissions : by setting users as operator (or “op” for short). General information on op status can be found at <http://minecraft.gamepedia.com/Op>

The abilities of op permission may be adjusted by altering the `op-permission-level` setting in the `../server/getting-started/configuration/server-properties` file.

A list of native Minecraft server commands available to players with op can be found at the [Minecraft Wiki](#).

Avertissement : Minecraft does not have any fine-grained permissions capacity, only op. This is a very high level of permission and should be reserved for trusted players. More complicated permission setups require the use of a permissions plugin or mod. Sponge is not a permissions-management plugin.

Note : Some plugins and mods may also grant specific permissions to ops.

Installing Plugins

What Are Plugins

Plugins are files written in Java that change the way the server works, generally adding features. They allow server owners to do a lot of modifications, such as adding an economy system, managing teleports and permissions, etc.

Finding Plugins

Avertissement : Do not download plugins from untrustworthy sources ! Malicious plugins can be used to give others unauthorized access to your server or computer.

SpongePowered currently runs the [Ore platform](#) to make it easy for plugin developers and users to distribute and download plugins. Alternatively you can search for plugins on the [SpongePowered forums](#).

Installation

SpongeForge

In order to install plugins, place them into the `/mods/` folder of your game or server directory. If your download came in a `.zip` file, then you may need to extract it to find a `.jar` file inside.

You can also place your plugins inside the `/mods/plugins/` subfolder or even set a custom folder in the `global.conf` file via the `plugins-dir` setting. SpongeForge will automatically search these folders for plugins. Please note that plugins which make use of Mixins *must* reside inside the `mods` folder.

SpongeVanilla

Note : For consistency between SpongeForge and SpongeVanilla, plugins are stored in the `mods` directory on SpongeVanilla.

In order to install plugins, place them into the `/mods/` folder of your game or server directory. If your download came in a `.zip` file, then you may need to extract it to find a `.jar` file inside.

You can also place your plugins inside the `/mods/plugins/` subfolder or even set a custom folder in the `global.conf` file via the `plugins-dir` setting. SpongeVanilla will automatically search these folders for plugins. Please note that plugins which make use of Mixins *must* reside inside the `mods` folder.

Common Problems

If you are having problems with a plugin, consider the following things :

- Is the plugin compatible with your Minecraft version ? While Sponge tries to keep old plugins working, this is sometimes not possible. In most cases, plugins based on a stable release should continue functioning without being updated.
- Do you run a current Java version ? Sponge is built for Java 8 and is known to trigger a bug in the JRE 1.8.0_40, so make sure you use a version newer than that.
- The plugin may be outdated. Is there a newer version of the plugin ?

- Does the plugin need a specific Implementation and/or Build of Sponge ? Some plugins may bypass the Sponge API, or otherwise rely on details that change between versions or platforms. Check at the site you downloaded it from.

Exploit Patches

In recent Sponge builds (SpongeForge 974+), SpongeForge and SpongeVanilla patch a few client-server exploits. Whenever the implementations detect a user performing an exploit, they are kicked from the server with a message explaining why they were kicked. If enabled, a log message is also sent to the console. More exploit patches may be added in the future.

Note : If you know about an exploit we currently *don't* cover, please let us know ! You can contact us via exploits@spongepowered.org or PM a staff member on the forums. Please **DO NOT** post exploits publicly on IRC, our GitHub repos or the forums, if they're still unknown. This prevents abuse until we get the issues fixed.

Exploits Patched implemented in Sponge

1. Sign command exploit where a client could run a command such as 'op'
2. Client could force the server to make the user respawn invisible
3. Client could set an itemstack's display name and cause it to exceed the character limit

Note that these patches can't be disabled, only the logging is configurable as of now.

Avertissement : The invisibility exploit patch has been disabled in recent Sponge builds due to the detection method falsely accusing users of performing the exploit.

Log Message Control

Log messages for the exploit patches can be individually controlled in the Sponge config file. Please read the [../getting-started/configuration/sponge-conf](#) page for more information. Here's a short overview of available options :

```
# Log when server receives exploited packet with itemstack name exceeding string_
↳limit.
exploit-itemstack-name-overflow=false

# Log when player attempts to respawn invisible to surrounding players.
exploit-respawn-invisibility=false

# Log when server receives exploited packet to update a sign containing commands from_
↳player with no permission.
exploit-sign-command-updates=false
```

Astuce : Log messages can also be controlled via a command, instead of directly editing the config file. For example, to enable the sign command exploit logging, type `sponge config -g logging.exploit-sign-command-updates true` in the console (You can also type the commands in-game if you are an op).

Performance Tweaks

Sponge provides several performance enhancing and tweaking options to help you run a lag-free server, even under heavy load.

Note : While we try to improve the performance of every server, the performance gains depend on your setup. Please run some tests to ensure you configured your Sponge server to get the most out of it !

Entity Activation Range

This setting will alter the loading behaviour of entities around players. Lowering the value will only load close entities, while raising it will also load entities that are far away from the player. Lower this to improve your servers performance, especially with high entity and player counts.

Astuce : It's possible to specify the activation range *per mob*. You can set `auto-populate` to `true` and Sponge will then add all available mobs to the activation range list, it is advised to disable it after the list is filled. If you add new mobs to the game, just repeat the procedure and those new mobs will also be added to the list below.

```
entity-activation-range {
    # If enabled, newly discovered entities will be added to this config with
    # a default value.
    auto-populate=false

    # Default activation ranges used for all entities unless overridden.
    defaults {
        ambient=32
        aquatic=32
        creature=32
        misc=16
        monster=32
    }
}
```

Async Lighting

This setting will run lighting checks on a separate thread to improve performance.

```
optimizations {
    # Runs lighting updates async.
    async-lighting=true
}
```

Cache Tameable Owners

This setting will cache tameable entities owners' UUID to save constant lookups from the data watcher.

```
optimizations {
    # Caches tameable entities owners to avoid constant lookups against data watchers.
    ↪ If mods cause issue, disable.
```

```
cache-tameable-owners=true
}
```

Drops Pre Merge

This setting will pre-process and potentially merge item drops to avoid spawning extra entities that are then merged post-spawning.

```
optimizations {
    # If enabled, block item drops are pre-processed to avoid
    # having to spawn extra entities that will be merged post spawning.
    # Usually, Sponge is smart enough to determine when to attempt an item pre-merge
    # and when not to, however, in certain cases, some mods rely on items not being
    # pre-merged and actually spawned, in which case, the items will flow right_
    ↪through
    # without being merged.
    drops-pre-merge=true
}
```

Auto-Saving Interval Adjustment

Vanilla Minecraft defaults to saving all chunks every 900 ticks (45 seconds). If you wish to raise or lower this interval, then change it in the servers `global.conf` file :

```
world {
    # The auto-save tick interval used when saving global player data.
    # Set to 0 to disable. (Default: 900) Note: 20 ticks is equivalent to 1 second.
    auto-player-save-interval=900

    # The auto-save tick interval used to save all loaded chunks in a world.
    # Set to 0 to disable. (Default: 900) Note: 20 ticks is equivalent to 1 second.
    auto-save-interval=900
}
```

Reducing this interval increases the load on your server's CPU and storage, but reduces the data loss that might occur if the server locks up or the power fails. Conversely, increasing the auto-save interval reduces the load on the hardware, but at the expense of increasing the amount of in-game progress that could be lost in case of server failure.

Becoming an Expert Spongineer

Contents

Commands

Commands are one method in which server operators can administer their server, and in which players can interact with the server.

In Sponge, commands follow a system of permissions. Permissions allow server operators to control who can access what commands. By default, all commands are granted to players with OP status. Players without operator status do not have access to administrative commands or commands that require an assigned permission node. A server operator can fine-tune who can access what commands by adding/negating permission nodes through a permissions plugin.

Note : Sponge is not a permissions-management plugin. To add and negate permissions for individual players or groups, you will need to find a permissions-management plugin.

Operator Commands

These commands, in addition to regular player commands, are available to server operators.

Sponge

The following commands are available to players with operator status (or the correct permission node) on servers powered by Sponge.

Command	Description	Permission
/sponge audit	Forces loading of unloaded classes to enable mixin debugging.	sponge.command.audit
/sponge chunks	Prints out the chunk data for a world, a dimension, or globally.	sponge.command.chunks
/sponge config	Alters a global, world, or a dimension config.	sponge.command.config
/sponge heap	Dumps the JVM heap.	sponge.command.heap
/sponge plugins	Lists currently installed plugins.	sponge.command.plugins
/sponge plugins reload	Asks plugins to perform their own reload procedures.	sponge.command.plugins.reload
/sponge reload	Reloads the global, world, or dimension config.	sponge.command.reload
/sponge save	Saves the global, world, or dimension config.	sponge.command.save
/sponge timings	The main command for the timings module.	sponge.command.timings
/sponge tps	Display ticks per second for each world.	sponge.command.tps
/sponge version	Prints the Sponge/SpongeAPI versions to the console.	sponge.command.version

Sponge Command Parameters

- /sponge chunks [-g] [-d dim] [-w world]
- /sponge config [-g] [-d dim] [-w world] key value
- /sponge save [-g] [-d dim|*] [-w world|*]
- /sponge reload [-g] [-d dim|*] [-w world|*]

Note : The `/sponge audit` command forces loading of any classes which have not yet been loaded, allowing the full output from all mixin debugging environment variables to be captured. This also requires the `mixins.checks` variable, see the [Mixin wiki](#) for more information.

Astuce : Here are a few simple examples of the `sponge config` command in action. Please see `../getting-started/configuration/index` for a more detailed explanation.

```
1. /sponge config logging.chunk-load true
```

Since no dimension was specified, the dimension would default to the sender(player) dimension. So if you were in a `mystcraft` dimension, this would alter the `mystcraft` dimension config.

2. `/sponge config -d nether logging.chunk-load true`

Since a dimension type was specified, this would alter the nether dimension config (and hence all nether worlds).

3. `/sponge config -w DIM1 logging.chunk-load true`

This would alter the config of world named DIM1.

Timings

Timings are a tool built into Sponge that allows server administrators to monitor the performance of their server. Timings will collect information about a server so that a report may later be generated on the data. Information that is recorded by timings include the server motd, version, uptime, memory, installed plugins, tps, percent of tps loss, amount of players, tile entities, entities, and chunks. Below is a list of sub-commands to `/sponge timings`:

Command	Description
<code>/sponge timings on</code>	Enables timings. Note that this will also reset timings data.
<code>/sponge timings off</code>	Disables timings. Note that most timings commands will not function and timings will not be recorded if timings are disabled.
<code>/sponge timings reset</code>	Resets all timing data and begins recording timing data after the time this command was done.
<code>/sponge timings report</code>	Generates the timings report on your server performance at https://timings.aikar.co
<code>/sponge timings verbon</code>	Enables timings monitoring at the verbose level.
<code>/sponge timings verboff</code>	Disables timings monitoring at the verbose level. Note that high-frequency timings will not be available.
<code>/sponge timings cost</code>	Gets the cost of using timings.

Forge

The following commands are available only when using the SpongeForge coremod on Forge. Other implementations of the Sponge API, such as SpongeVanilla, do not include these commands.

Command	Description	Permission
<code>/forge tps</code>	Display ticks per second for each world.	<code>forge.command.forge</code>
<code>/forge track</code>	Enable tile entity tracking.	<code>forge.command.forge</code>

For any Forge mods that use the vanilla command API, command permissions are provided in the form `<modid> . command . <commandname>`.

Vanilla

There are several commands built-in to vanilla Minecraft that are also available on servers powered by Sponge. The list below is not comprehensive, but it includes the most commonly used commands. These commands are available to players with operator status (or the correct permission node). In general, permissions for vanilla Minecraft commands on a Sponge server are of the structure `minecraft . command . <command>`, as shown below.

Command	Description	Permission
/ban	Ban a player.	minecraft.command.ban
/ban-ip	Ban a player's IP address.	minecraft.command.ban-ip
/banlist	View all banned players.	minecraft.command.banlist
/clear	Clear an inventory.	minecraft.command.clear
/deop	Remove OP from a player.	minecraft.command.deop
/difficulty	Set the game difficulty.	minecraft.command.difficulty
/gamemode	Set the gamemode of a player.	minecraft.command.gamemode
/gamerule	Set a gamerule.	minecraft.command.gamerule
/give	Give an item to a player.	minecraft.command.give
/kill	Kill a player or entity.	minecraft.command.kill
/op	Give Operator status to a player.	minecraft.command.op
/pardon	Remove a player from the ban list.	minecraft.command.pardon
/save-all	Save the server.	minecraft.command.save-all
/save-off	Disable automatic server saving.	minecraft.command.save-off
/save-on	Enable automatic server saving.	minecraft.command.save-on
/setidletimeout	Define how long players can be idle before getting kicked.	minecraft.command.setidletimeout
/setworld-spawn	Set the spawnpoint for the world.	minecraft.command.setworldspawn
/stop	Stop the server.	minecraft.command.stop
/toggedown-fall	Toggle between sunny and rainy weather.	minecraft.command.toggedownfall
/tp	Teleport players and entities.	minecraft.command.tp
/weather	Set the weather to a defined condition.	minecraft.command.weather
/whitelist	Manage the server whitelist.	minecraft.command.whitelist
/worldborder	Manage the world border.	minecraft.command.worldborder

Sponge also creates a spawn-protection bypass permission, and two permissions for controlling the ability to edit commandblocks. Note that these permissions use the actual *name* of the commandblock, which is normally @ by default.

- Allow player to bypass spawn-protection on all worlds : `minecraft.spawn-protection.override`
- Allow editing an ordinary commandblock of the given name : `minecraft.commandblock.edit.block.<name>`
- Allow editing a minecart commandblock of the given name : `minecraft.commandblock.edit.minecart.<name>`

Player Commands

The following commands are available as part of vanilla Minecraft to players without operator status.

Command	Description	Permission
/help	View information on commands used on the server	minecraft.command.help
/me	Tell everyone what you are doing.	minecraft.command.me
/say	Display a message to everyone (or, if using selectors, specific players).	minecraft.command.say
/tell	Privately message another player.	minecraft.command.tell

A full list of vanilla commands can be found at : http://minecraft.gamepedia.com/Commands#List_of_commands. Permissions for vanilla Minecraft commands on a Sponge server are of the structure `minecraft.command.<command>`.

Troubleshooting

You're probably here because something went wrong with your Sponge server. Let's see if we can figure out what it was, and what to do about it.

Potential Sources of Trouble

- *Java Is Not Installed On Your Computer*
- *Network Connection Failure (or DDoS Attack)*
- *Not Enough Free Memory*
- *Malformed Config File (eg. Bad Editing)*
- *A Plugin (or Mod) Has Malfunctioned*
- *Operating System Unstable (eg. Virus Infection)*
- *Corrupted Data*
- *Problem Between Keyboard and Chair*
- *There is a Bug in Sponge*
- *There is Something Wrong With the Universe*

Java Is Not Installed On Your Computer

Solution : Get Java. Visit the `../server/getting-started/jre` for more information.

Network Connection Failure (or DDoS Attack)

Symptoms : Network connection is very laggy, drops in and out, or absent.

Solutions : Check your connection to the modem or router. See if your browser has similar troubles. You can use a free service like speedtest.net to check your connection speed. Other services running on your computer or local network may be the cause. Make sure that you have enabled Port Forwarding on your router. A DDoS attack, while unlikely, will probably completely kill your connectivity, and you should contact your ISP if you believe this to be the case.

Not Enough Free Memory

Symptoms : Server crashes, often accompanied with "Out of Memory" messages.

Solutions : Expand the maximum Perm memory size with the startup argument `-XX:MaxPermSize=128`. Expand your server heap memory (if possible) with startup arguments eg. `-Xms1024M` (1GB starting memory) and `-Xmx2048M` (2GB maximum). Monitor your free memory on the computer and see if there is some locked up in other processes. You may need to kill frozen java processes, or restart your machine. Memory leaks sometimes occur with bugs in plugins, which can take time to isolate.

Still an issue ? : If you are still having issues despite the above and cannot increase the Heap Size, check in your Task Manager to see if you are using all available Memory. If you are, the only solution is to add more RAM to your system. If there is still plenty of memory available, you are running 32-bit Java. If you are using 32-bit Java, we recommend an upgrade to 64-bit Java, provided that your Operating System is also 64-bit.

Malformed Config File (eg. Bad Editing)

Symptom : One (or more) plugins refuse to load, or behave in unexpected ways. The server log files will contain messages about unreadable files on startup. The server may crash, and data may be corrupted.

Solution : Stop the server, and check your edited files. Load backup files of any corrupted data. You may need to delete a config file entirely and allow it to regenerate upon server startup.

A Plugin (or Mod) Has Malfunctioned

Symptom : This could be almost anything - whatever your plugins do, plus the X-factor. Commonly the server crashes with a train of error messages in the server log files.

Solution : Stop the server, and check to see nothing has been corrupted. Be sure to check that it isn't from an incorrectly edited config file (above). Remove suspect plugins and add them again one by one, restarting the server each time. The problem may originate from one plugin that is out of date - check for updates. Plugin conflict may also be the cause, having two incompatible plugins.

Operating System Unstable (eg. Virus Infection)

Symptom : The server keeps crashing or timing out, and other parts of your operating system are also having problems.

Solution : Stop everything. Thoroughly check your system and storage devices for malware and viruses. Good tools for this include AdwCleaner, Junkware Removal Tool, MalwareBytes, and most antivirus ware. Check your server files for corruption after a clean restart of your system. Examine the hardware for damage too if the problems persist - eg. a faulty power supply.

Corrupted Data

Symptom : World files fail to load or cause server to crash when players enter certain chunks. Database corruption.

Solution : Load backup files of corrupted data. Software for repairing damaged worlds is available, and missing regions may be regenerated. Investigate the cause of corruption - was it a malformed plugin, database driver, power failure or something else ? Always make sure you make regular backups of important data onto a secure device.

Problem Between Keyboard and Chair

Symptom : Everything was working fine yesterday. It went strange today after I did XYZ ...

Solution : SpongeDocs is not large enough to encompass the things people may do that will cause software to fail in unpredictable ways. It is always worth thinking long and hard about what you may have done recently that could have affected the smooth running of your server. A memory card may be loose after dusting, a shortcut may be broken...

There is a Bug in Sponge

Symptom : None of the above apply, and it still doesn't work as it should.

Solution : Time to get out the big guns. File a report on the [SpongeForge](#) or [SpongeVanilla](#) issue tracker, remembering to include details of the version of Forge and Sponge you are using, and a link to the relevant server log file.

There is Something Wrong With the Universe

We can't help you with this one. You're on your own.

Log Files

Logfiles are an essential part when it comes to debugging your server and figuring what went wrong. This pages contains logfiles from SpongeForge and SpongeVanilla servers including short descriptions.

Provided Logfiles

- *SpongeForge logfiles*
 - *fml-junk-earlystartup.log*
 - *fml-server-latest.log*
 - *latest.log*
- *SpongeVanilla logfiles*
 - *latest.log*
- *Reading logfiles*
- *Common errors*

SpongeForge logfiles

SpongeForge writes several logfiles to the `/logs` folder located inside your servers directory. As of Forge 1521 these are :

1. `fml-junk-earlystartup.log`
2. `fml-server-latest.log`
3. `latest.log`

fml-junk-earlystartup.log

Note : Only a few example lines are shown here. To read the full example log, follow this link : [SpongeForge 1521 fml-junk-earlystartup.log file](#)

fml-server-latest.log

Note : Only a few example lines are shown here. To read the full example log, follow this link : [SpongeForge 1521 fml-server-latest.log](#)

```
[main/INFO] [FML/]: Forge Mod Loader version 11.14.3.1521 for Minecraft 1.8 loading
[main/INFO] [FML/]: Java is Java HotSpot(TM) 64-Bit Server VM, version 1.8.0_51,
↳ running on Windows 8.1:amd64:6.3, installed at ##PATH_TO_JAVA_HERE##
[main/DEBUG] [FML/]: Java classpath at launch is forge.jar
[main/DEBUG] [FML/]: Java library path at launch is ##PATH_TO_JAVA_HERE##
```


The example log indicates that we're running :

- Forge 11.14.3.1521 (Version 1521)
- Java 8 64bit Update 51
- Windows 8.1 x64
- the directory Java was installed to (Line 4)

Avertissement : SpongeForge won't run on Java 6 (1.6.x) or Java 7 (1.7.x). If you encounter an error stating that you run an older Java build than Java 8, please update your JRE to 1.8.x and try again !

```
[main/DEBUG] [FML/]: Examining for coremod candidacy spongeforge-1.8-1521-2.1-DEV-750.  
↪ jar  
[main/INFO] [FML/]: Loading tweaker org.spongepowered.asm.launch.MixinTweaker from ↪  
↪ spongeforge-1.8-1521-2.1-DEV-750.jar
```

This indicates that SpongeForge 750 was found and loaded by Forge. For further help regarding the SpongeForge naming scheme, have a look here : [../getting-started/implementations/spongeforge/](#).

latest.log

Note : Only a few example lines are shown here. To read the full example log, follow this link : [SpongeForge 1521 latest.log](#)

This is the output that you would see in the Minecraft server GUI.

SpongeVanilla logfiles

latest.log

Note : Only a few example lines are shown here. To read the full example log, follow this link : [SpongeVanilla 47 latest.log](#)

This is the output that you would see in the Minecraft server GUI.

Reading logfiles

If you're unsure on how to read a common crashlog, you'll find help here, but first we need a crashlog. For this short introduction we will just use an example crash from the [Debugging](#) page : Example crashlog of an outdated SpongeForge build.

```
WARNING: coremods are present:  
SpongeCoremod (sponge-1.8-1499-2.1DEV-575.jar)  
Contact their authors BEFORE contacting forge
```

The first thing you'll notice is a Warning that coremods are present. Nothing to worry about here, that's not an error, just a warning to contact Sponge support, not Forge.

```
java.lang.NoClassDefFoundError: org/spongepowered/api/event/game/state/
↳GameStartingServerEvent
```

A few lines below the actual error is found. In this case it's a `NoClassDefFoundError`. If you're unsure what that means, head over to our [Debugging](#) page. If it's a common error, it will be listed there. If it isn't, you can always ask on the forums for help! Make sure you provide the full crashlog.

Luckily your systems details are included at the bottom of the crashlog :

```
Minecraft Version: 1.8
Operating System: Windows 8.1 (amd64) version 6.3
Java Version: 1.8.0_51, Oracle Corporation
Java VM Version: Java HotSpot(TM) 64-Bit Server VM (mixed mode), Oracle Corporation
Memory: 515666256 bytes (491 MB) / 782761984 bytes (746 MB) up to 1847590912 bytes
↳ (1762 MB)
JVM Flags: 0 total;
IntCache: cache: 0, tcache: 0, allocated: 0, tallocated: 0
FML: MCP v9.10 FML v8.0.99.99 Minecraft Forge 11.14.3.1521 5 mods loaded, 5 mods
↳ active
States: 'U' = Unloaded 'L' = Loaded 'C' = Constructed 'H' = Pre-initialized 'I' =
↳ Initialized 'J' = Post-initialized 'A' = Available 'D' = Disabled 'E' = Errored
UC   mcp{9.05} [Minecraft Coder Pack] (minecraft.jar)
UC   FML{8.0.99.99} [Forge Mod Loader] (forge.jar)
UC   Forge{11.14.3.1521} [Minecraft Forge] (forge.jar)
UC   Sponge{1.8-1499-2.1DEV-575} [SpongeForge] (minecraft.jar)
U     Core{unknown} [Core Plugin] (Core.jar)
Loaded coremods (and transformers):
SpongeCoremod (sponge-1.8-1499-2.1DEV-575.jar)
```

This indicates that

- Minecraft 1.8 with Forge 1521 was running on
- Java 8 Update 51 (64bit version) and that
- 2 additional mods were installed
 - SpongeForge 1.8-1499-2.1DEV-575 (which is build #575) and
 - Core

Note : Please note that the other three installed mods (mcp, FML, Forge) are required on every Forge server and necessary to run properly.

Now the following assumptions can be made :

- maybe the plugin crashed the server
- SpongeForge doesn't match the Forge version : 1499 required, 1521 installed

If you want to know how to solve this, head over to our checklist on the [Debugging](#) page.

Common errors

Head over to [Debugging](#) to read about common errors and exceptions.

Debugging

Logs are an essential part when it comes to debugging your server and figuring what went wrong. This page will show some basic logging examples and will try to explain what you can do to fix your issues, when encountering them.

Checklist

Whenever you encounter a crash or warning make sure you set SpongeForge or SpongeVanilla up correctly. Here's a short checklist to help you out. If you're unsure on how to acquire the information needed, have a look at the [Log Files](#) page. It explains how you get the desired answers out of your logfiles.

1. Is Java 8 installed and is Sponge using it ?

Sponge requires Java 8 and will crash when using Java 7 or older.

2. Is the recommended Forge version installed ?

Usually SpongeForge will run on older or newer Forge builds than the recommended build. However it is strongly advised to run the recommended build only. If you encounter a crash and your versions are mismatching, match them first and try again. If you're unsure which Forge build you need, take a look at [/server/getting-started/implementations/spongeforge](#)

3. Are there any other coremods (besides SpongeForge) installed ?

Some coremods modify Forge in a way that makes it impossible to run SpongeForge properly. If you have coremods installed and Sponge crashes, try to remove them and test again. Please report any incompatible Coremods on [GitHub](#) or the [Sponge Forums](#). This allows staff to solve these issues as soon as possible.

4. Is every plugin you're using built against your desired Sponge build ?

The Sponge API is subject to change sometimes. When you try to use an older plugin on the most recent Sponge build and a crash occurs, try downgrading Sponge or contact the plugin author to get an updated plugin. If you're on an older Sponge build and a recent plugin crashes, try to update Sponge first. If that doesn't fix the issue, contact the plugin author and ask for a fix.

5. Separating a faulty plugin

If the problem still persists, try to remove all plugins and re-add them one by one while trying to start the server every time you added a plugin.

If you're still unsure why and what exactly crashed, have a look at your crashlog. Some common crashes and common solutions are listed below.

General Warnings

A common source of errors and bugs is a version mismatch between either SpongeForge and Forge or SpongeForge and Plugins. First we'll have a look at the general warning Forge gives us upon crashing :

```
WARNING: coremods are present:
    SpongeForge (sponge-1.8-1521-2.1DEV-750.jar)
Contact their authors BEFORE contacting forge
```

This isn't a bug or error, it's just Forge telling you that a Coremod (here : SpongeForge) is installed. Forge advises you to contact the Sponge developers first, before asking the Forge support for help. Nothing to worry about.

Common Exceptions

Here are some common exceptions and some reasons why you might encounter them.

Note : If you encountered a crash, error or any other malfunction **not** listed here, please report it on the [Sponge Forums](#) or on [GitHub](#). This will help others, who are running into the same issue.

Mismatched SpongeForge and Forge

```
[12:59:21] [main/ERROR] [mixin/]: @Mixin target net.minecraftforge.event.world.  
↳BlockEvent$NeighborNotifyEvent was not found mixins.forge.core.json:event.block.  
↳MixinEventNotifyNeighborBlock
```

This is a common crash when you try to run SpongeForge on the wrong Forge build. Note that the target/Mixin can vary. Always match Forge against SpongeForge ! If you're unsure which version of Forge is required and you already got your SpongeForge build, take a look at : [../getting-started/implementations/spongeforge/](#)

Other common errors

```
Caused by: java.lang.ClassNotFoundException: org.spongepowered.api.event.state.  
↳ServerStartedEvent  
Caused by: java.lang.NullPointerException
```

The first error indicates that a Class is missing, the second is a NullPointerException which indicated that the plugin you're trying to use relies on missing parameters. This happens when you try to run an older plugin on a newer SpongeForge or SpongeVanilla build and vice versa.

```
java.lang.AbstractMethodError: net.minecraft.entity.player.EntityPlayerMP.  
↳getTabList()Lorg.spongepowered/api/entity/living/player/tab/TabList;  
at (...)
```

An `AbstractMethodError` occurs when a plugin tries to call a method which isn't implemented yet. Please check if you're running the most current build of Sponge and update if a newer version is available. If the problem still exists, either report it on the official [Issuetracker](#), on the forums or on IRC. You can request the implementation of the missing feature too.

```
[Server thread/INFO]: Starting minecraft server version 1.8  
[Server thread/ERROR]: Encountered an unexpected exception  
java.lang.NoClassDefFoundError: org.spongepowered/api/event/game/state/  
↳GameStartingServerEvent
```

Note : Read the full example crashlog here : [SpongeForge 575 crashlog with a plugin built against build 750](#)

A `NoClassDefFoundError` occurs when the plugin tries to access a class that isn't on the classpath. This happens when the API got adjusted or refactored lately and you're trying to run an older plugin on a newer build of Sponge and vice versa. Always try to use the correct version ! Either ask the Plugin author which Sponge version he build against or try updating/downgrading your SpongeForge or SpongeVanilla to solve this.

Reporting Bugs

If you've encountered a bug and you're unsure on how to report it, this is the page to look at. We're currently handling all bug reports through our issue trackers on GitHub and via the forums.

If there's an issue with SpongeForge, report it on the [SpongeForge issue tracker](#). The same applies for SpongeVanilla ([SpongeVanilla issue tracker](#)) and even the Docs ([SpongeDocs issue tracker](#)).

Whenever you report a bug, please include the following :

- Sponge version used (SpongeForge or SpongeVanilla build number)

- Forge version (if applicable)
- any other mods or plugins installed including their versions
- your *Log Files*
- please use a service like [GitHubs gist](#) or [pastebin](#) to provide logs, *don't* paste them into the post directly
- your crash report (optional, not necessarily created)

Please make sure you already read through our *Debugging*, *Troubleshooting* and *Log Files* sections. If the problem still persists, then file a bug report.

Here's a short example, you may copy and use it when opening a ticket on GitHub. Note that this is just a suggestion, its usage is optional and you may modify it to suit your needs :

```
Build number: #buildnumber here#
Forge version: #Forgeversion here#

Log files: #link to pastebin here#
Crash report: #link to pastebin here#

Description:

#short description of the bug you found here#
```